IN THE CLAIMS

Please amend the claims as follows:

1. (Previously Presented) A method of executing a plurality of threads within a single programmable processor, the method comprising:

storing a plurality of data elements in partitioned fields of at least one register having a register width, each of the data elements having an elemental width smaller than the register width;

receiving an instruction stream for each one of the plurality of threads at an execution unit; and

executing instructions from each instruction stream received at the execution unit in a multistage pipeline such that, at a given time, the multistage pipeline includes instructions from different ones of the instruction streams in different stages of the multistage pipeline, the instructions including a single instruction that specifies an operation to cause multiple instances of the operation to be performed, each instance of the operation to be performed using a different one of the plurality of data elements in partitioned fields of the at least one register to produce a catenated result.

2. (original) The method of claim 1 wherein the number of threads executing within the execution unit is prime relative to a rate of execution of a slowest functional unit in the execution unit.

3. (original) The method of claim 1 wherein the instructions from the plurality of instruction streams are executed in a round-robin manner.

4. (Previously Presented) The method of claim 1 wherein only one thread from the plurality of threads can handle an exception at a given time.

5. (original) The method of claim 1 further comprising:

decoding a second single instruction specifying a third and a fourth register each containing a plurality of floating-point operands;

multiplying the plurality of floating point operands in the third register by the plurality of operands in the fourth register to produce a plurality or products; and

providing the plurality of products to partitioned fields of a result register as a catenated result.

6. (currently amended) A computer-readable storage medium:

~~having~~ encoded with instructions including an instruction stream for each one of a plurality of threads that instruct a computer system to perform operations comprising,

storing a plurality of data elements in partitioned fields of at least one register having a register width, each of the data elements having an elemental width smaller than the register width;

receiving an instruction stream for each one of the plurality of threads at an execution unit;

executing instructions from each instruction stream received at the execution unit in a multistage pipeline such that, at a given time, the multistage pipeline includes instructions from different ones of the instruction streams in different stages of the multistage pipeline, the instructions including a single instruction that specifies an operation to cause multiple instances of the operation to be performed, each instance of the operation to be performed using a different one of the plurality of data elements in partitioned fields of the at least one register to produce a catenated result.

7. (previously presented) The computer-readable storage medium of claim 6 wherein the number of threads executing within the execution unit is prime relative to a rate of execution of a slowest functional unit in the execution unit.

8. (previously presented) The computer-readable storage medium of claim 6 wherein the instructions from the plurality of instruction streams are executed in a round-robin manner.

9. (Previously Presented) The computer-readable storage medium of claim 6 wherein only one thread from the plurality of threads can handle an exception at a given time.

10. - 14. (canceled)

15. (Previously Presented) The computer-readable storage medium of claim 6 wherein the computer system is to perform operations further comprising:

decoding a second single instruction specifying a third and a fourth register each containing a plurality of floating-point operands;

multiplying the plurality of floating point operands in the third register by the plurality of operands in the fourth register to produce a plurality or products; and

providing the plurality of products to partitioned fields of a result register as a catenated result.